AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Amended) A method for defeating a denial-of-service attack, for use in a communication system in which a client sends a ciphertext of a random number chosen by the client encrypted under a public key of a server to authenticate ~~a~~ the server, the method comprising the steps of:

(a) at the server, generating a random number r_B in response to a service request from a client and sending the random number to the client;

(b) at the server, receiving ~~the~~ a ciphertext produced by the client using the random number r_B from the ~~client~~ server and a random number r_A ~~of~~ selected by the client, enciphered with the public key of the server;

(c) at the server, recovering a random number ~~r_B~~ $r_{B'}$ from the ciphertext received from the client based on a private key corresponding to the public key of the server and comparing the recovered random number with the random number sent to the client; and

(d) if the recovered random number is equal to the random number sent to the client ~~numbers match at the step (c)~~, providing the service, and, otherwise, denying the service.

2. (Amended) The method as received in claim 1, wherein, at the step (a), the random number r_B is obtained by an equation $r_B = H(K_{master}, index\_r_B)$ where H is a hash function, K_{master} is a secret master key and $index\_r_B$ is an index parameter for the random number.

3. (Amended) A method for defeating denial-of-service attack, applicable to a server authentication system in which a client uses a discrete exponentiation g^{r_A} as a ~~random~~ client's challenge to a ~~the~~ server, a private key and a public key of ~~a~~ the server are respectively b and g^b, and the ciphertext of the client's challenge using the public key of the server is g^{br_A}, the method comprising the steps of:

(a) at the server, sending a random number r_A to a client;

(b) at the server, receiving x and y values which the client computed by using the random number from the server as:

$$x = (g^b)^{r_A + r_B}$$

where b is the private key of the server and g^b is the public key of the server, and

$$y = h(g^{r_A})$$

where h represents a hash function;

(c) comparing y from the client with y' as follows:

$$y' = h(x^{b^{-1}} g^{-r_B})\text{; and}$$

(d) if y and y' match, providing a requested service to the client, and, otherwise, denying the service to the client.

4. (Amended) In a communication system having a large capability processor in which a client sends a server a ciphertext of a random number encrypted under a public key of the server to authenticate the server, a computer readable medium for recording a program for implementing the functions of:

(a) at the server, generating a random number r_B in response to a service request from a client and sending the random number to the client;

(b) at the server, receiving ~~the~~ a ciphertext which is produced by the client based on the random number r_B sent to the client and a random number r_A ~~of~~ produced by the client;

(c) at the server, recovering ~~the~~ a random number ~~r_B~~ $r_{B'}$ from the ciphertext received from the client and comparing the recovered random number with the random number sent to the client; and

(d) if the recovered random number is equal to the random number sent to the client ~~numbers match at the step (c)~~, providing the service, and, otherwise, denying the service.

5. (Amended) In a server authentication system having a large capability processor, in which a client uses a ~~disrecte~~ discrete exponentiation g^{r_B} as a ~~random~~ client's challenge to a server, a private key and a corresponding public key of the server

are respectively b and g^b, and a ciphertext of the client's challenge using the public key of the server is g^{br_A}, a computer readable medium for recording a program for implementing the functions of:

(a) at the server, sending a random number to a client;

(b) at the server, receiving x and y values which the client computed by using the random number from the server as:

$$x = (g^b)^{r_A + r_B}$$

where b is the private key of the server and g^b is the public key of the server, and

$$y = h(g^{r_A})$$

where h represents a hash function;

(c) at the server, comparing y from the client with y' as follows:

$y' = h(x^{b^{-1}} g^{-r_B})$; and

(d) if y and y' match, providing a service to the client, and, otherwise, denying the service.